
Microsoft
WinHEC
2005
Microsoft
Windows
Hardware & Driver Central

Performance Analysis on Multi-Processor systems using AMD CodeAnalyst for Windows

Sherry Hurwitz
SW Applications Manager
SRD
Advanced Micro Devices

Lei Yu
Member Technical Staff
SRD
Advanced Micro Devices

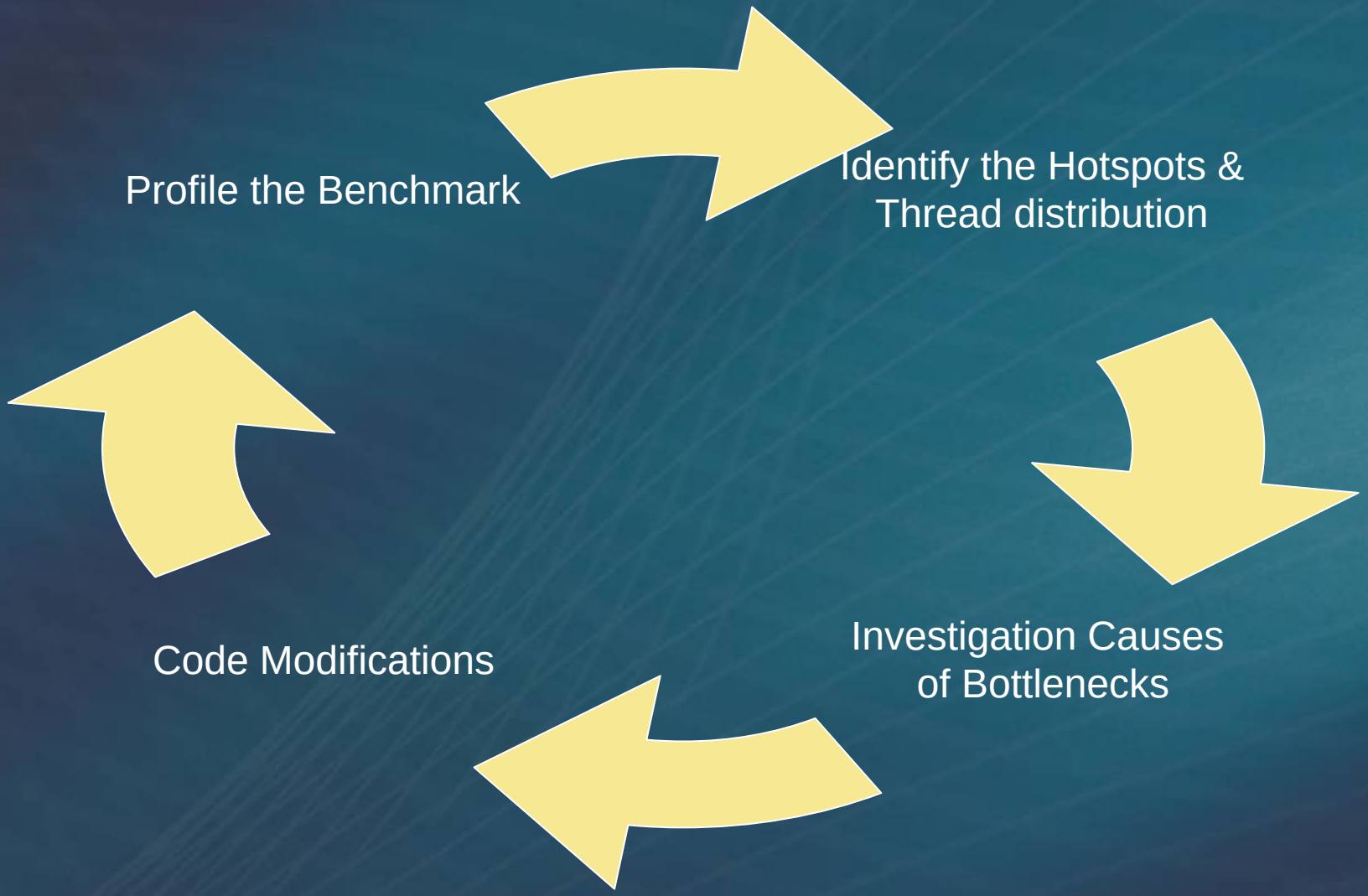
Session Outline

- Overview of Performance Analysis
  - Reasons for Poor Performance
  - The Optimization Process
- AMD CodeAnalyst Performance Analysis Tool
  - General Capabilities of CodeAnalyst
  - General Functionality of CodeAnalyst
  - Profile Capabilities
  - Thread Analysis
  - Pipeline Simulation
  - Partner Collaborations

Reasons for Poor Performance

- Inefficient algorithms
- Slow memory access
- High loop counts
- Branch prediction problem
- Slow instructions
- Object Synchronization

The Optimization Process



General Capabilities of AMD CodeAnalyst

- CodeAnalyst CAN:
  - Identify program bottlenecks
  - Monitor and Analyze software performance
  - Assist programmers in optimizing their software
- CodeAnalyst CANNOT:
  - Identify defects in your program

    (Tune your program after it runs fairly stable.)
- Environments CodeAnalyst runs on:
  - Windows: WinNT + SP4 up, Win2K + SP3, WinXP + SP1, 64bits WinXP, Server 2003
  - Linux: Red Hat, SuSe (both 32-bit/64-bit) with kernel 2.4 up.

General Functionality of AMD CodeAnalyst

- Profiling
  - Timer-based sampling
  - Event-based sampling
- Thread analysis
- Execution Pipeline Simulation

Profile Capabilities

- Low overhead system-wide profile
- Timer-based profile: On APIC enabled system, the finest time resolution is 0.1 ms and 1.0ms APIC disabled system.
- Event-based profile: designed to profile the 32 public events of AMD Athlon and AthlonXP, and 78 performance events on AMD Opteron and Athlon 64.
- CodeAnalyst can profile up to 4 performance events simultaneously.
- CodeAnalyst can profile on multiple processor system – up to an 8 CPU system.

Useful events: Unhalted CPU Clock Cycle

- Event 0x76, BU “Unhalted CPU Clock Cycle”, increases on every CPU clock cycle if system does not halt.
- The event could be very useful if the target application has negligible I/O activity.

For instance, benchmark Business Winstone 2004 has many GUI idles during each subtest. Profiling event 0x76 could provide more accurate bottleneck info.

* Exception: The event 0x76 counter increases even when system idles on Win64 XP.

Useful events: Memory access & Branching

Data Cache

- Event 0x40 Data Cache access
- Event 0x41 Data Cache miss
- Event 0x42 Data Cache refill from L2
- Event 0x43 Data Cache refill from system
- Event 0x44 Evicted Line
- Event 0x45 L1 DTLB miss and L2 DTLB hit
- Event 0x46 L1 and L2 DTLB miss
- Event 0x47 Misaligned data reference

Instruction Cache

- Event 0x80 ICache Fetch
- Event 0x81 ICache Miss
- Event 0x84 L1 ITLB miss and L2 ITLB hit
- Event 0x85 L1 ITLB miss and L2 ITLB miss

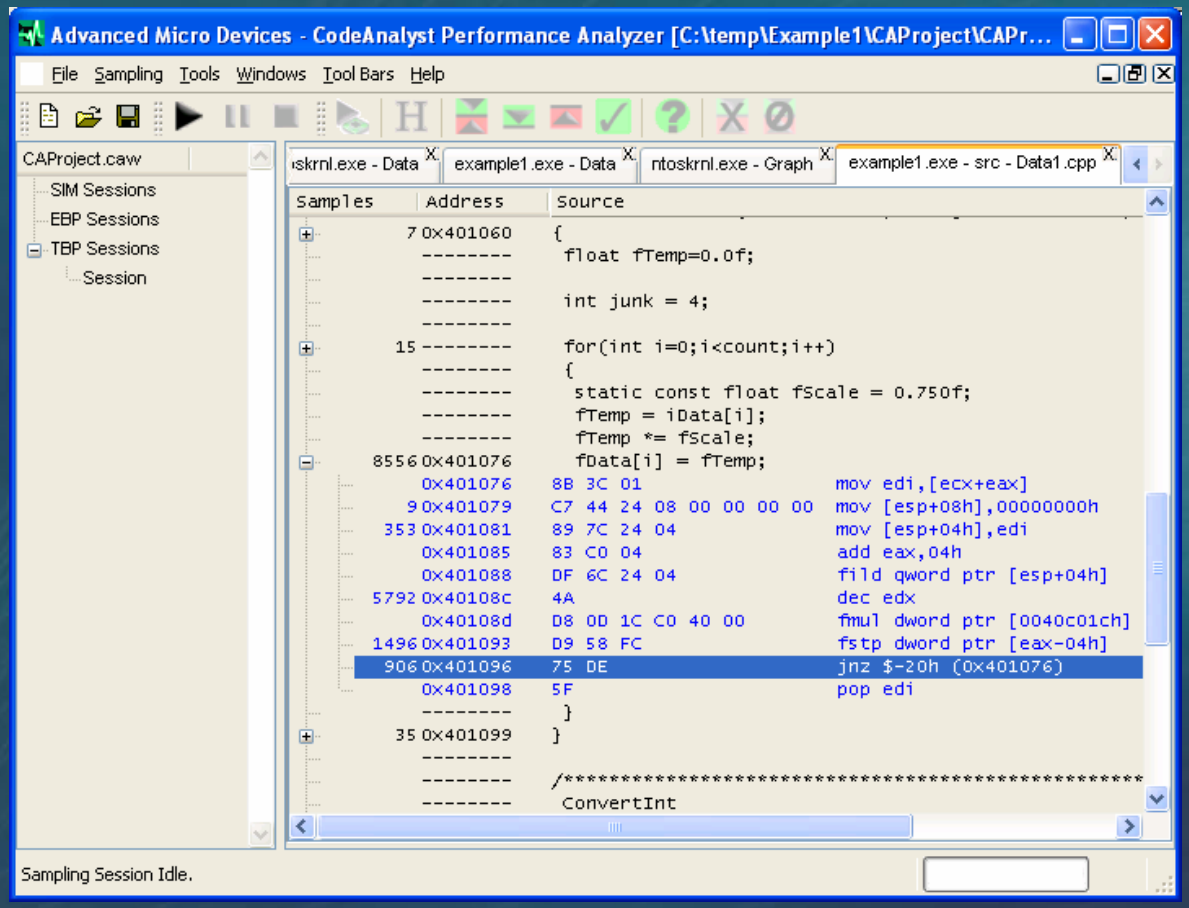
Branching (FR source)

- 0xC2 Retired Branches including exceptions and interrupts
- 0xC3 Retired branch mispredicted
- 0xC4 Retired taken branches
- 0xC5 Retired taken branches mispredicted

AMD CodeAnalyst – Data Views

- System Data View
- System Graph View
- Module Data View
- Module Graph View
- Source View
- Disassembly View

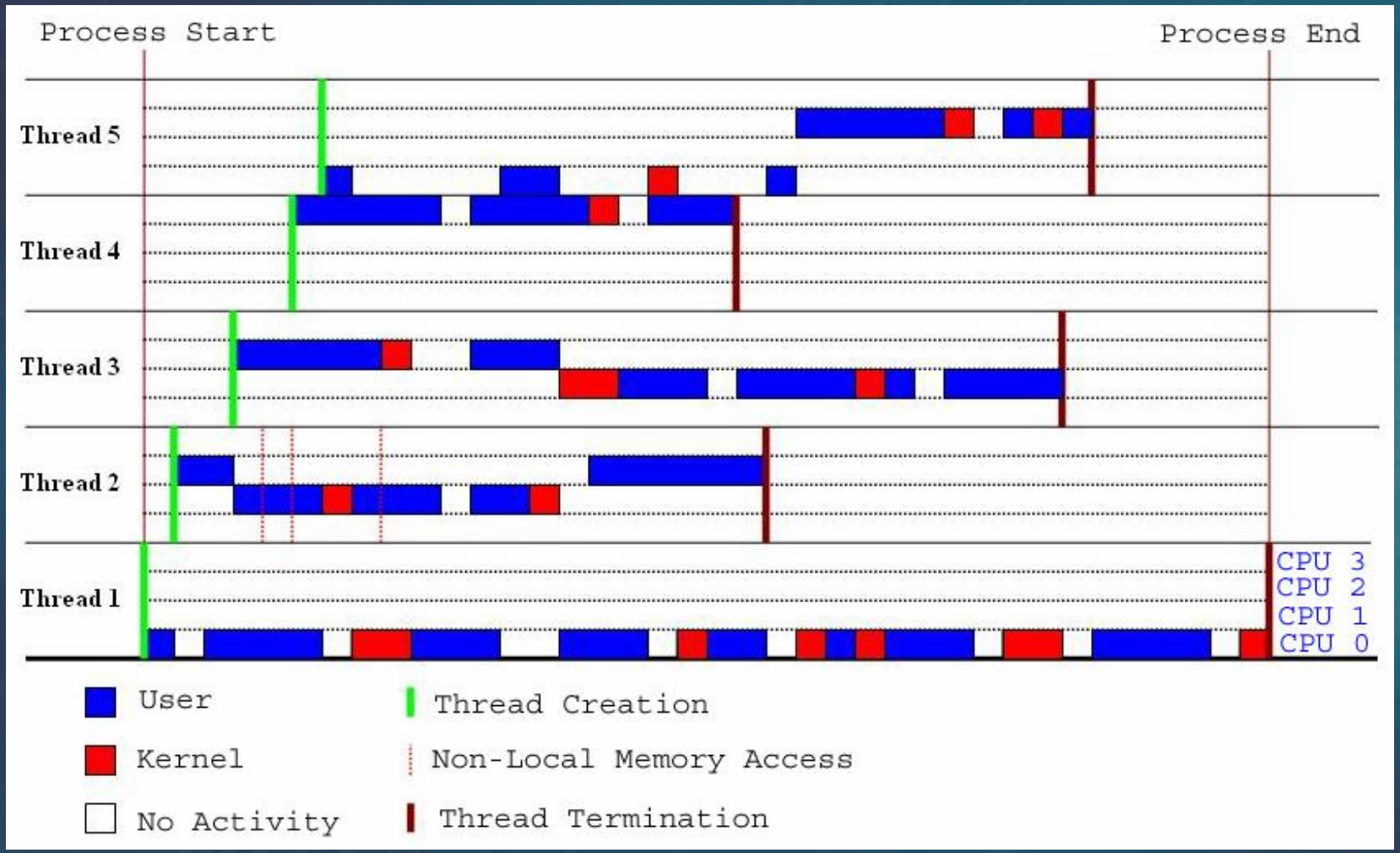
Demo will show the details of each of these views and the navigation between the views.



Thread Analysis

- Identities threads in the target application.
- Shows Thread creation and termination
- Monitors CPU affinity of each thread
- Identifies Non-local memory access
- Graphs thread activity on each CPU

Thread Analysis Data View



Pipeline Simulation

CodeAnalyst can simulate a block of code that the user specifies on different AMD microprocessors and provide cycle-precise execution info.

Requirement:

To define a code block to simulate, it requires the user to provide debug info for the target module.

Limitation:

- Cannot simulate instructions inside system space
- Cannot simulate multi-thread

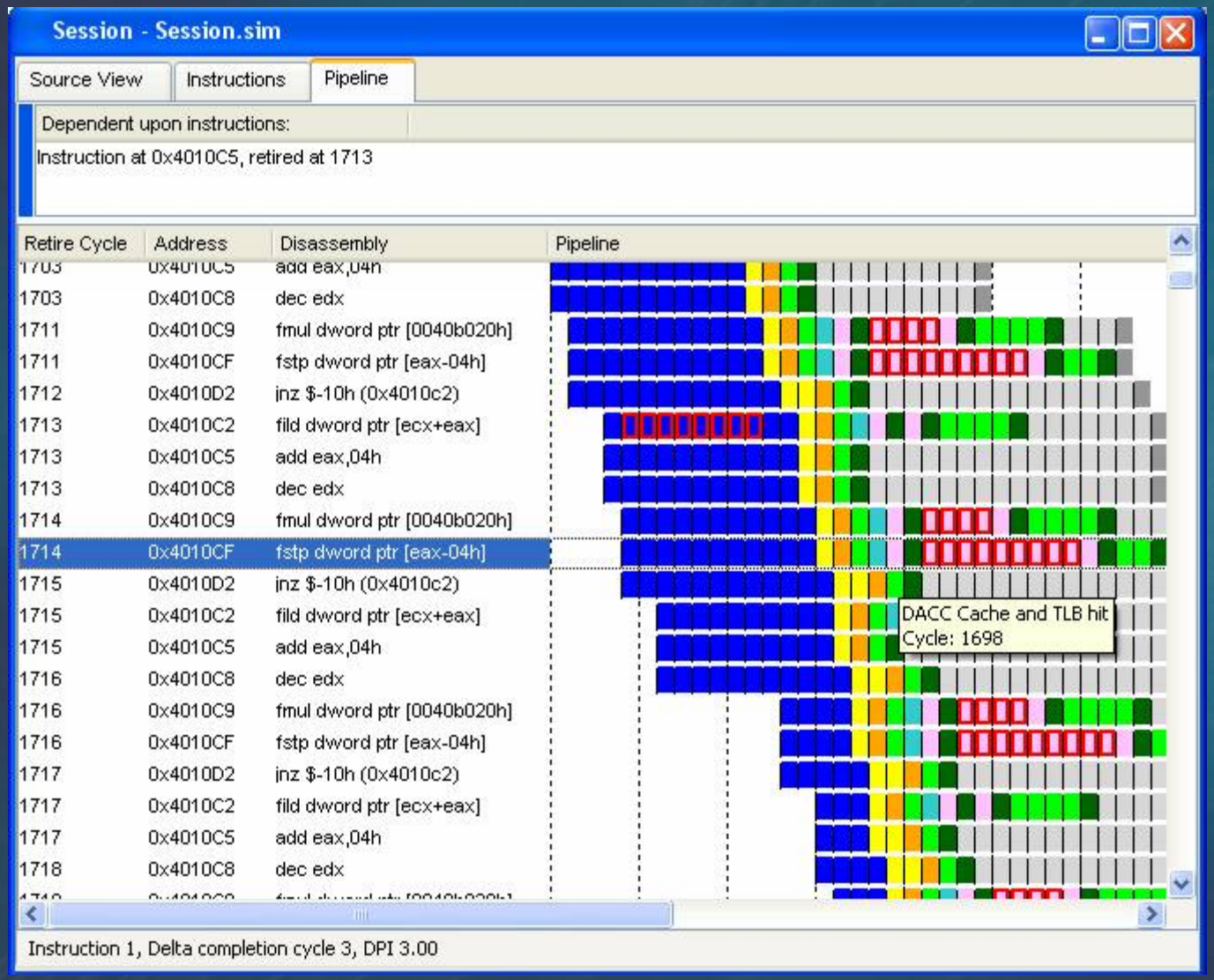
Some Assumptions in the Simulator

- Assumes perfect memory subsystem
  - All LSUops hit in the DCache
    - Assumes that 1 misaligned load = 2 back-to-back aligned loads (64-bit)
    - Assumes no cache bank conflicts
  - 100% Icache hit rate
- Assumes perfect branch prediction
- Assumes all schedulers are of infinite size

CodeAnalyst Simulation Demo

- Simulation configuration
- Trace point window
- Trace Point Start, Trace point End, trigger
- Pipeline view
  - Pipeline stage
  - Penalty
  - Dependency
  - Delta completion
  - IPC
- Simulation History

Pipeline Data View



Partner Collaborations

- Driver JIT code
  - data sample capture and interpretation
  - Special signature agreement with driver developer and CodeAnalyst
- DirectX vertex shader
  - Read the memory in JIT Address locations
  - Generate the PE dll from JIT code
- Leverage Java Virtual Machine Profiler Interface (JVMPI) for Java 2 SDK (1.4.x)
- Leverage Java Virtual Machine Tool Interface (JVMTI) for Java 2 SDK (1.5.x)
- Microsoft Common Language Runtime profile

Additional Resources

- Web Resources:
  - CodeAnalyst Download: http://www.amd.com/devcentral
  - Software Optimization Guide for AMD Athlon 64 and AMD Opteron: http://www.amd.com/devcentral